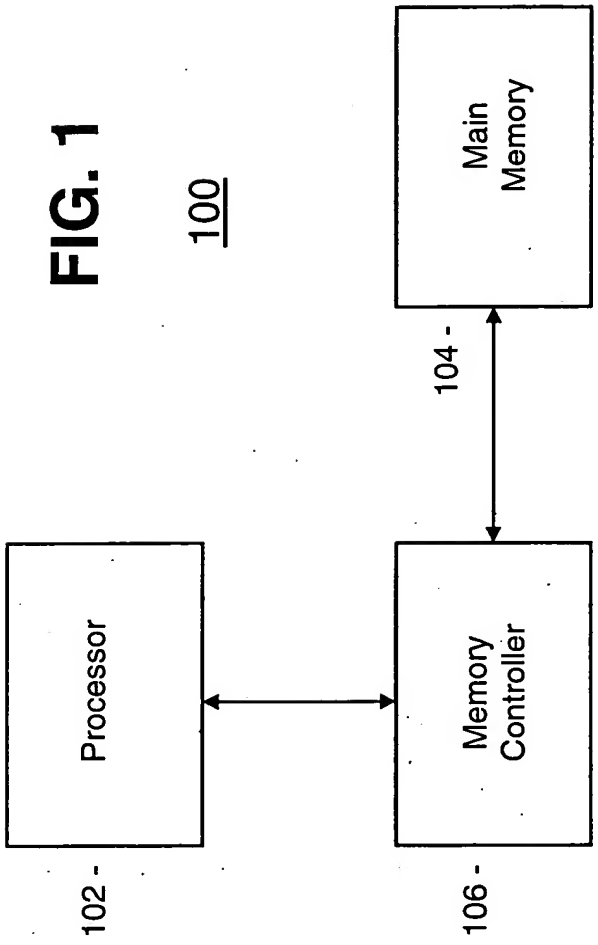
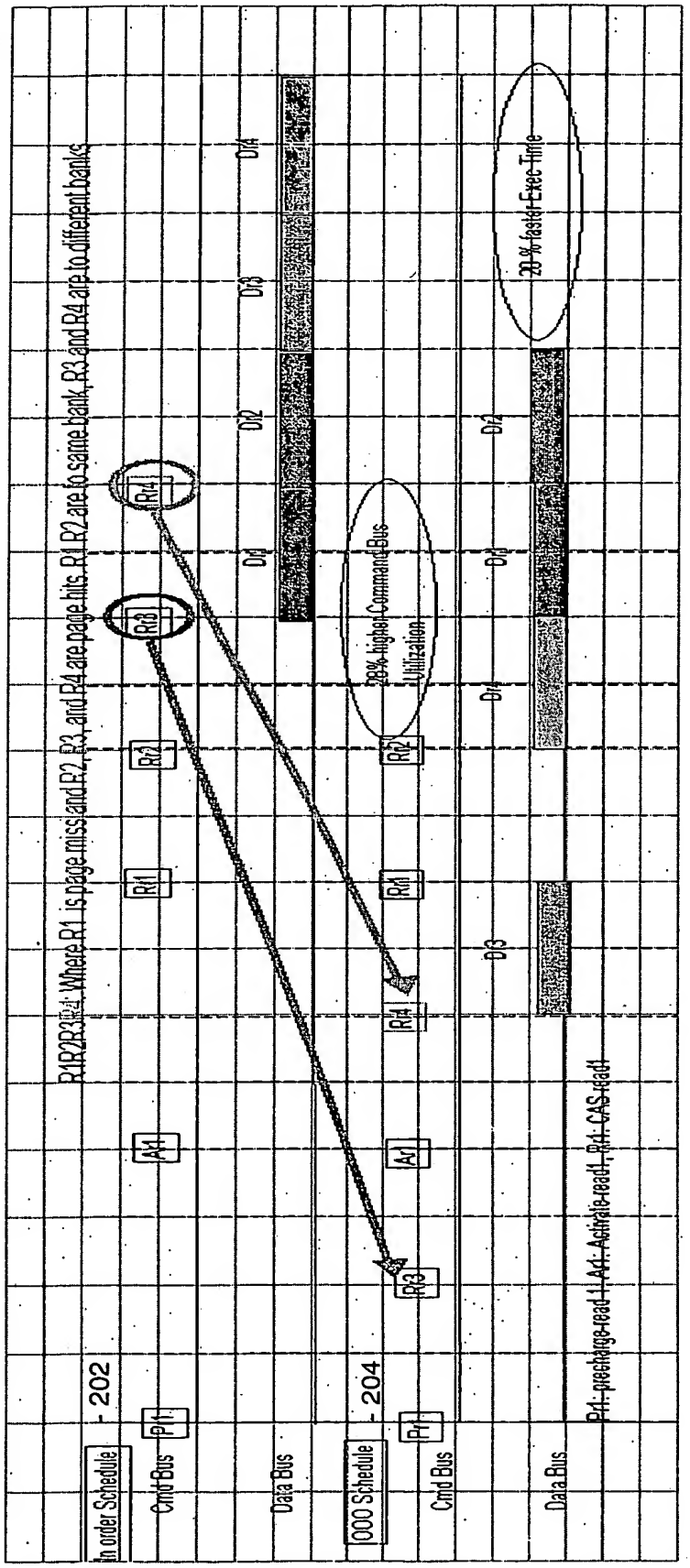
FIG. 1




R1R2R3R4: Where R1 is page miss and R2, R3, and R4 are page hits. R1 R2 are to same bank, R3 and R4 are to different banks
In order Schedule
202
Cmd Bus
P1
Data Bus
R1
R2
R3
R4
A1
D1
D2
D3
D4
28% higher Command Bus Utilization
OOO Schedule
204
Cmd Bus
Pr
Data Bus
Pr
D2
D1
Pr
D3
20 % faster Exec Time
P1: precharge-read 1, A1: Activate-read1, R1: CAS-read1

200

FIG. 2

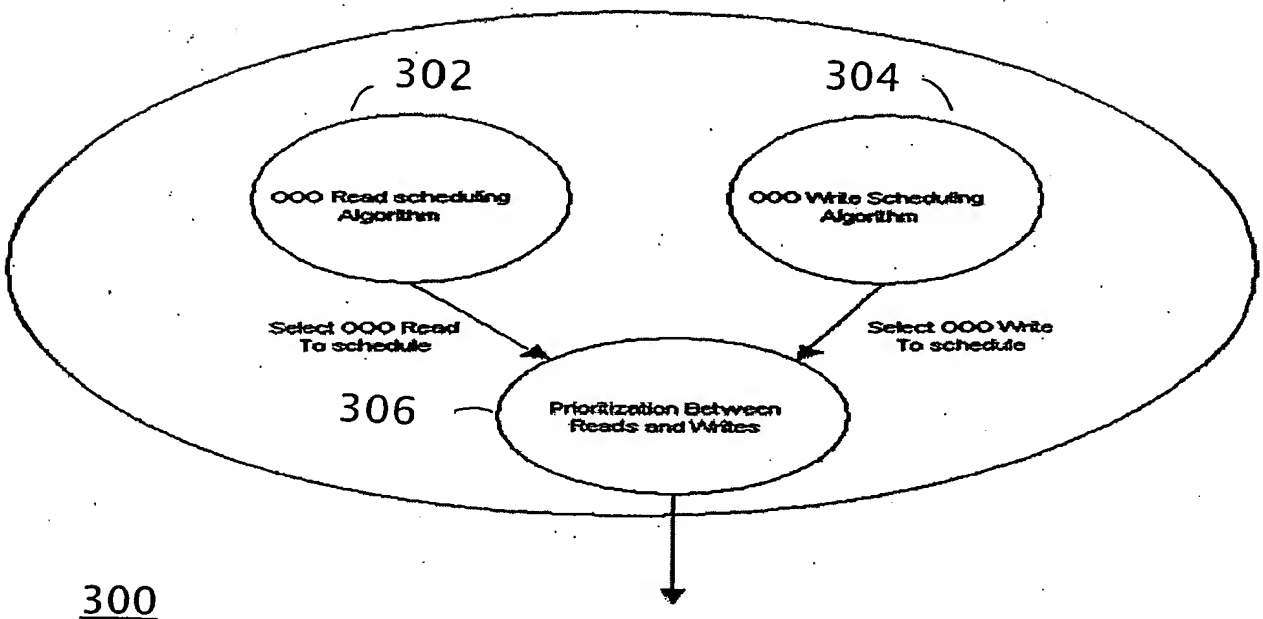

302
304
OOO Read scheduling Algorithm
OOO Write Scheduling Algorithm
Select OOO Read To schedule
Select OOO Write To schedule
306
Prioritization Between Reads and Writes
300

FIG. 3

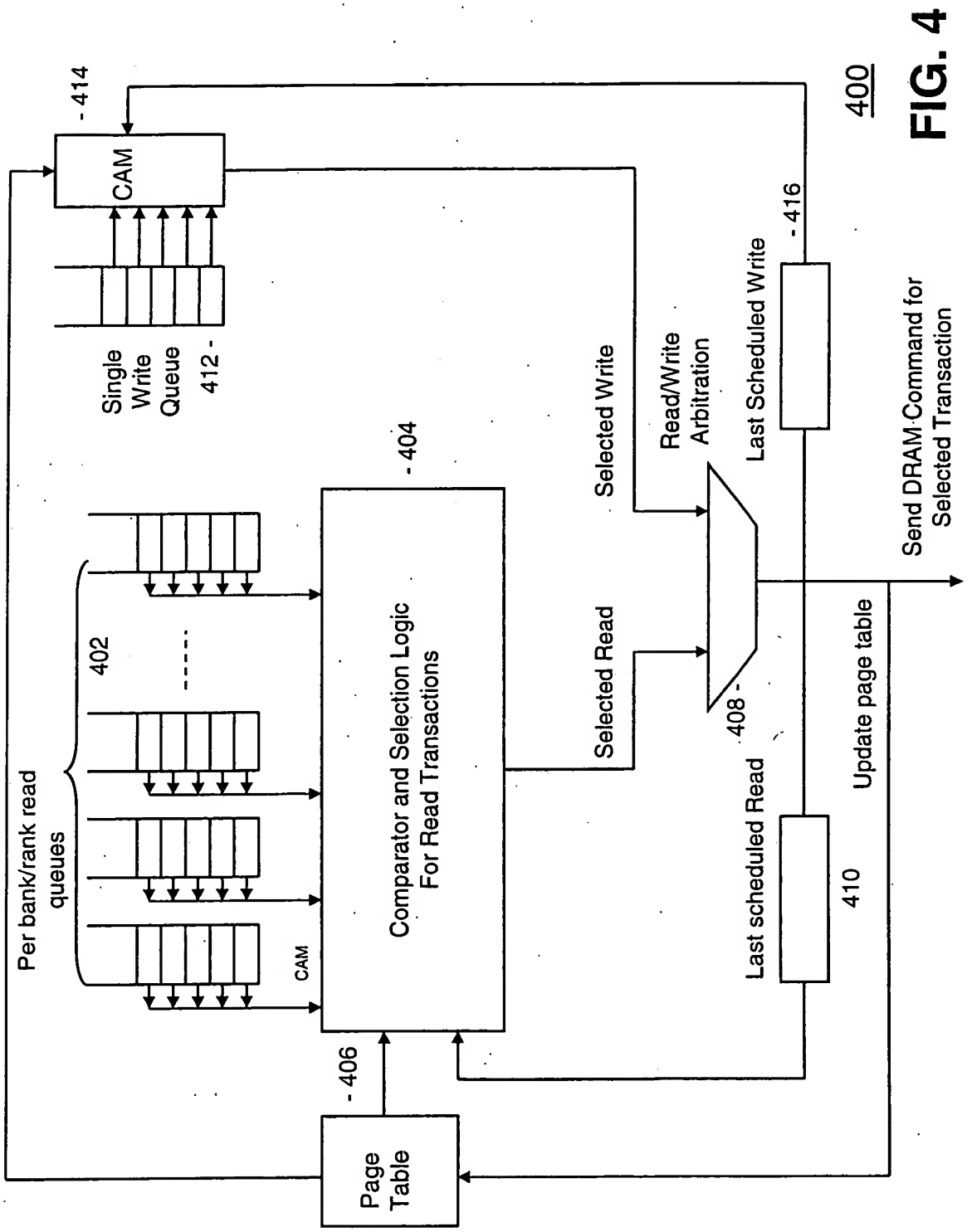

400
414
CAM
Single Write Queue
412
Per bank/rank read queues
402
CAM
Comparator and Selection Logic For Read Transactions
404
406
Page Table
Selected Write
Selected Read
Read/Write Arbitration
408
Last Scheduled Write
416
Last scheduled Read
410
Update page table
Send DRAM Command for Selected Transaction

FIG. 4

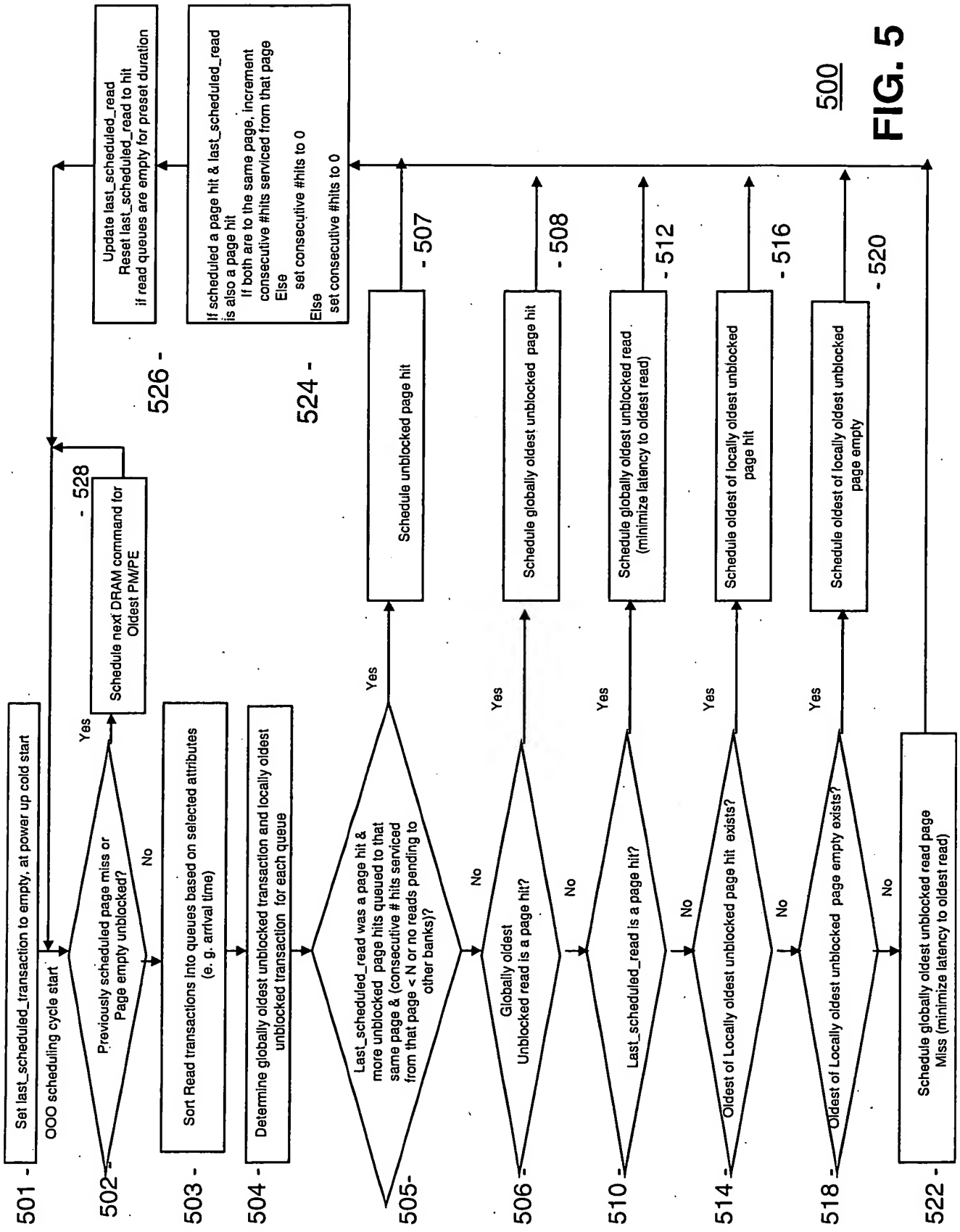

501 -
Set last_scheduled_transaction to empty, at power up cold start
OOO scheduling cycle start
502 -
Previously scheduled page miss or Page empty unblocked?
Yes
No
- 528
Schedule next DRAM command for Oldest PM/PE
503 -
Sort Read transactions into queues based on selected attributes (e. g. arrival time)
504 -
Determine globally oldest unblocked transaction and locally oldest unblocked transaction for each queue
505-
Last_scheduled_read was a page hit & more unblocked page hits queued to that same page & (consecutive # hits serviced from that page < N or no reads pending to other banks)?
Yes
No
- 507
Schedule unblocked page hit
506 -
Globally oldest Unblocked read is a page hit?
Yes
No
- 508
Schedule globally oldest unblocked page hit
510 -
Last_scheduled_read is a page hit?
Yes
No
- 512
Schedule globally oldest unblocked read (minimize latency to oldest read)
514 -
Oldest of Locally oldest unblocked page hit exists?
Yes
No
- 516
Schedule oldest of locally oldest unblocked page hit
518 -
Oldest of Locally oldest unblocked page empty exists?
Yes
No
- 520
Schedule oldest of locally oldest unblocked page empty
522 -
Schedule globally oldest unblocked read page Miss (minimize latency to oldest read)
524 -
If scheduled a page hit & last_scheduled_read is also a page hit
If both are to the same page, increment consecutive #hits serviced from that page
Else
set consecutive #hits to 0
Else
set consecutive #hits to 0
526 -
Update last_scheduled_read
Reset last_scheduled_read to hit
if read queues are empty for preset duration
500

FIG. 5

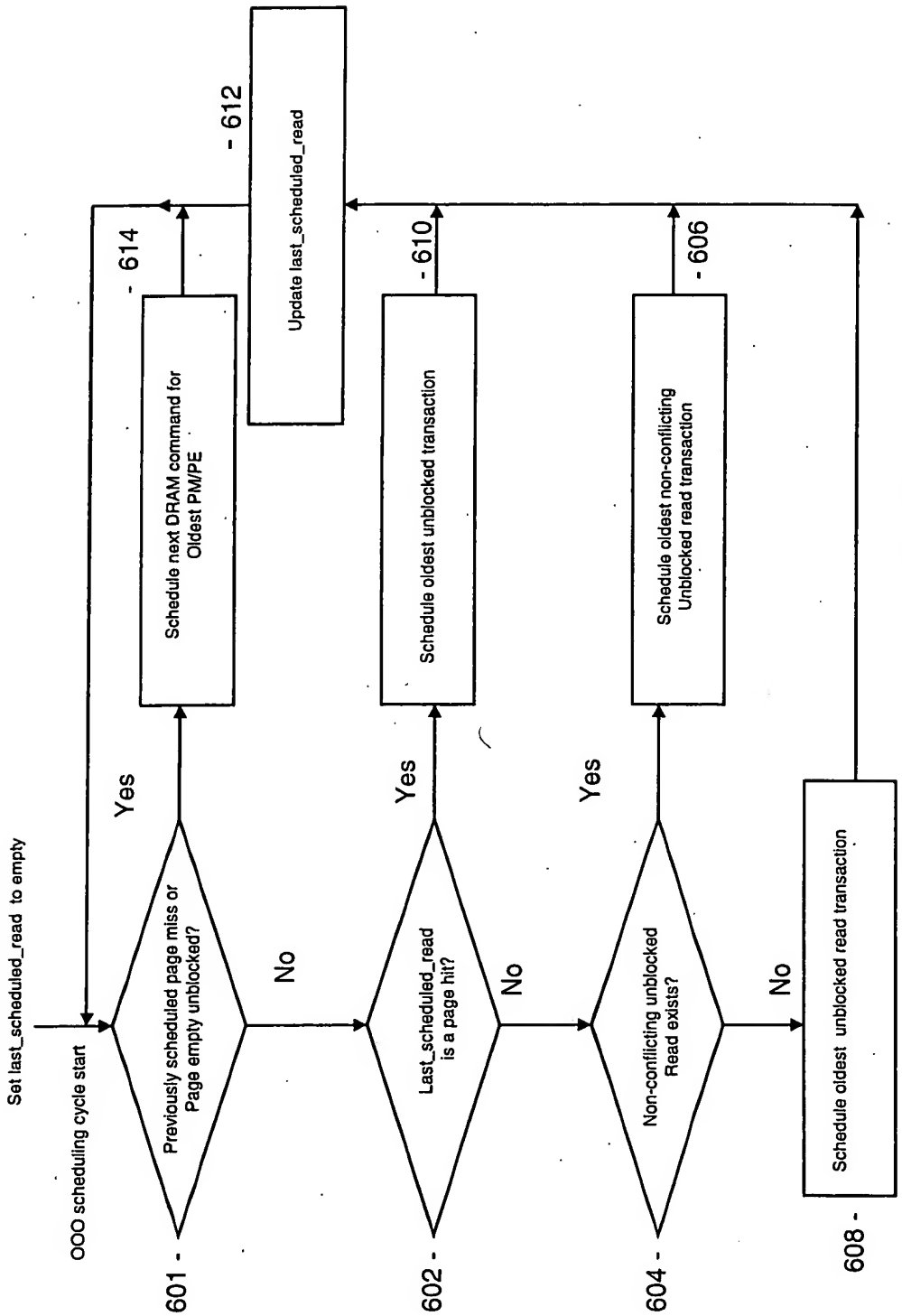

OOO scheduling cycle start
Set last_scheduled_read to empty
601 -
Previously scheduled page miss or Page empty unblocked?
Yes
No
602 -
Last_scheduled_read is a page hit?
Yes
No
604 -
Non-conflicting unblocked Read exists?
Yes
No
608 -
Schedule oldest unblocked read transaction
- 606
Schedule oldest non-conflicting Unblocked read transaction
- 610
Schedule oldest unblocked transaction
- 614
Schedule next DRAM command for Oldest PM/PE
- 612
Update last_scheduled_read

FIG. 6 600

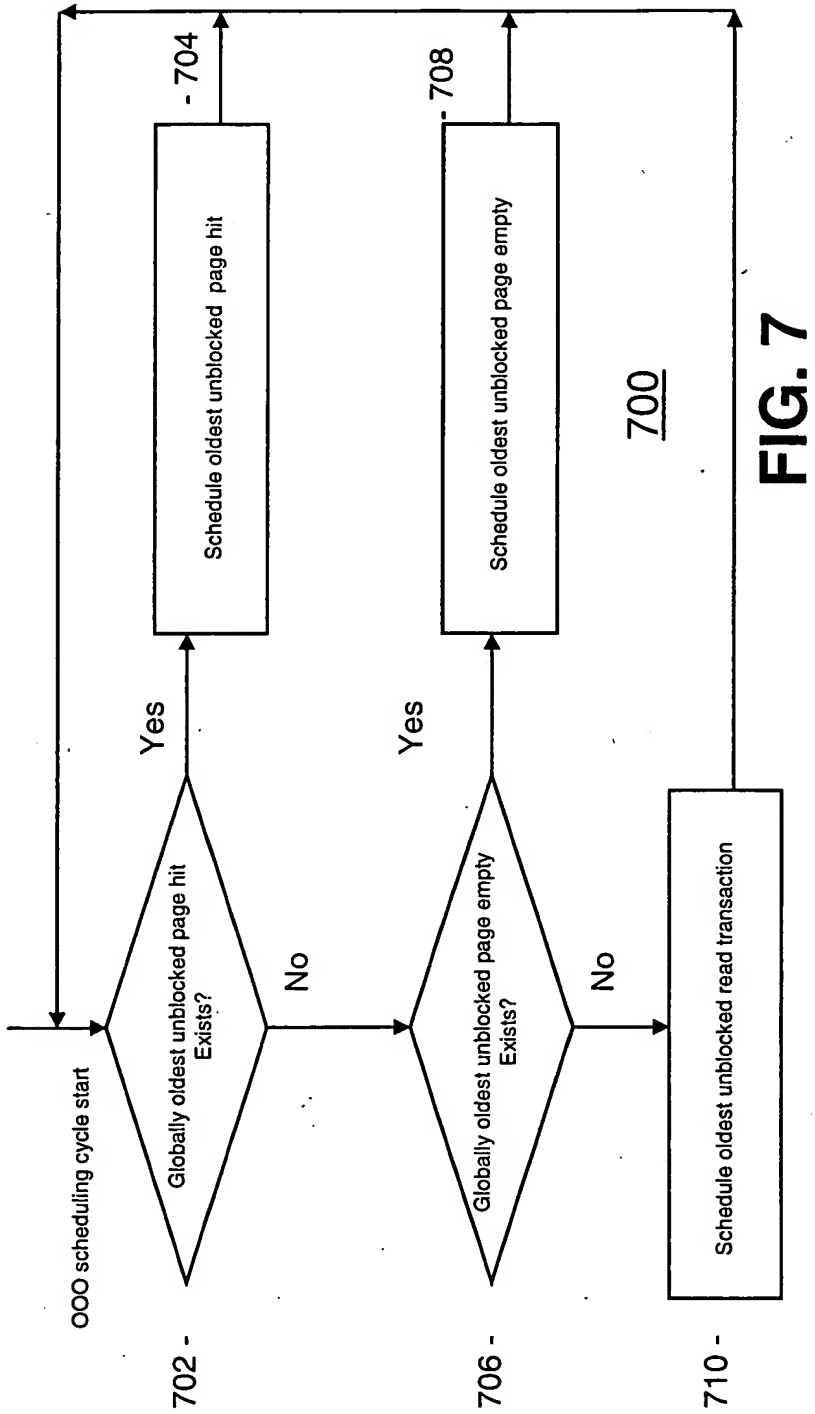

OOO scheduling cycle start
Globally oldest unblocked page hit Exists?
Yes
No
Schedule oldest unblocked page hit
704
Globally oldest unblocked page empty Exists?
Yes
No
Schedule oldest unblocked page empty
708
Schedule oldest unblocked read transaction
702
706
710
700

FIG. 7

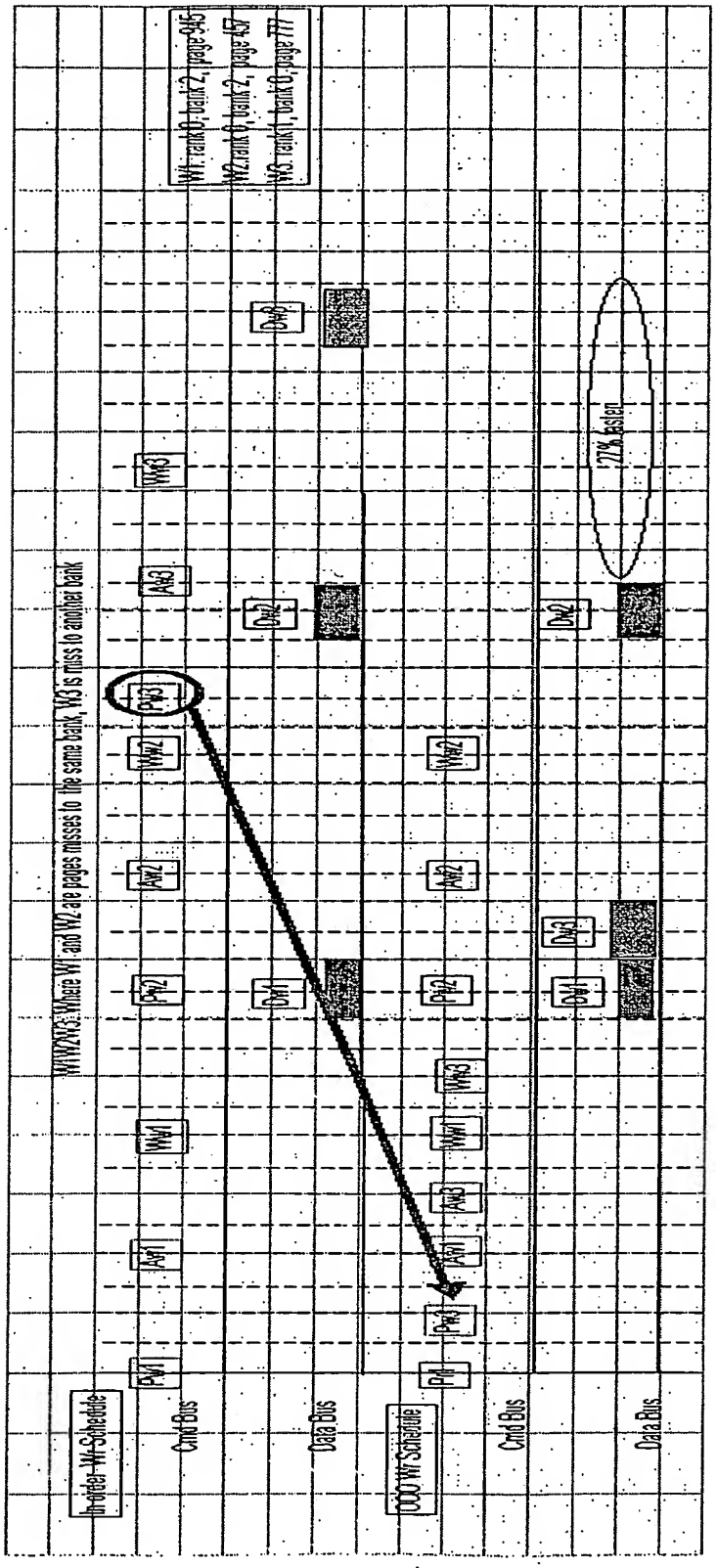

W1: rank0, bank2, page 345
W2: rank0, bank2, page 457
W3: rank1, bank0, page 777
W1W2NXE: Where W1 and W2 are pages misses to the same bank, W3 is miss to another bank
In order W1 Schedule
Cmd Bus
Data Bus
OOO W1 Schedule
Cmd Bus
Data Bus
27% faster
802
804

FIG. 8

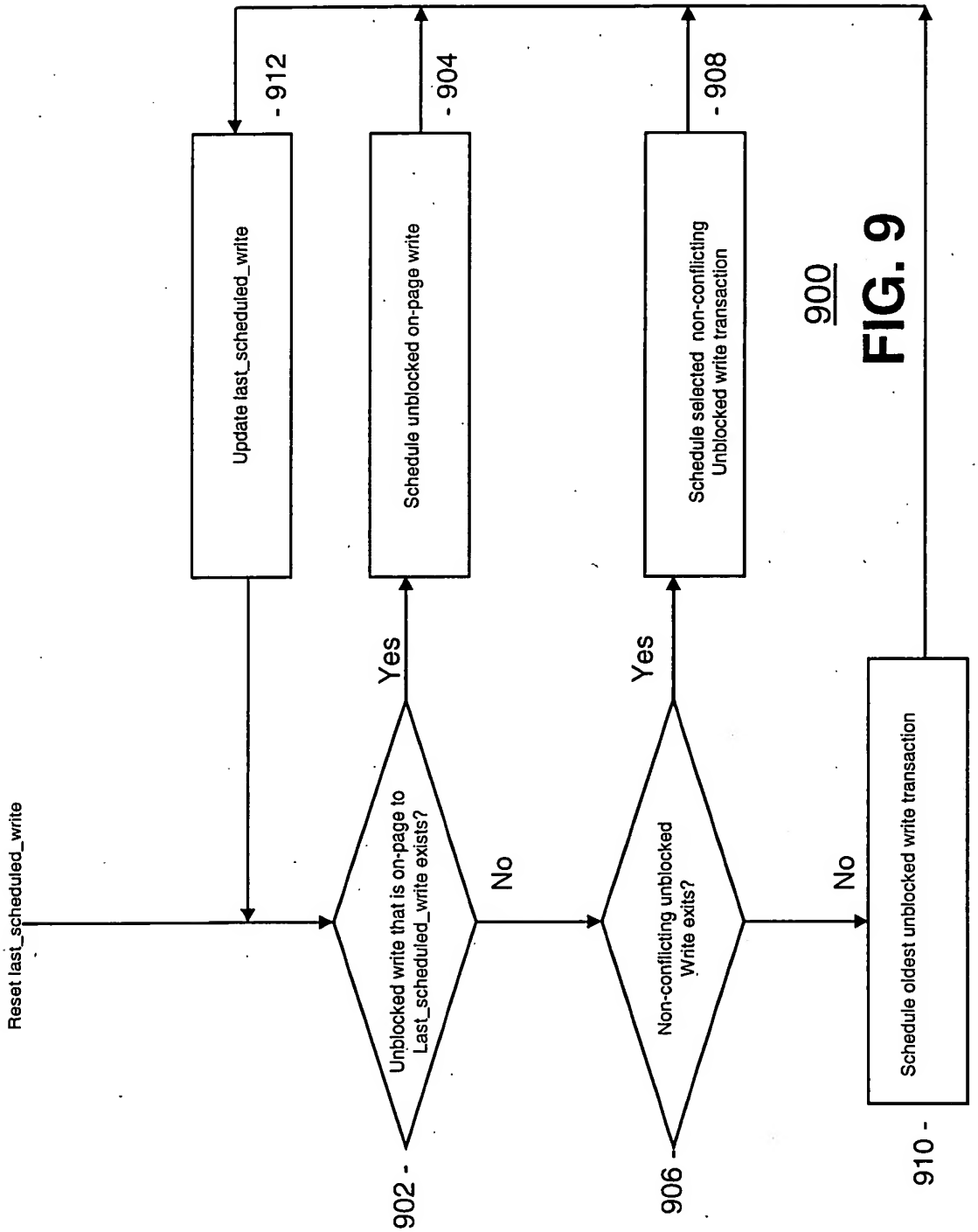

Reset last_scheduled_write
Unblocked write that is on-page to Last_scheduled_write exists?
902
Yes
No
Schedule unblocked on-page write
904
Non-conflicting unblocked Write exits?
906
Yes
No
Schedule selected non-conflicting Unblocked write transaction
908
Schedule oldest unblocked write transaction
910
Update last_scheduled_write
912
900

FIG. 9

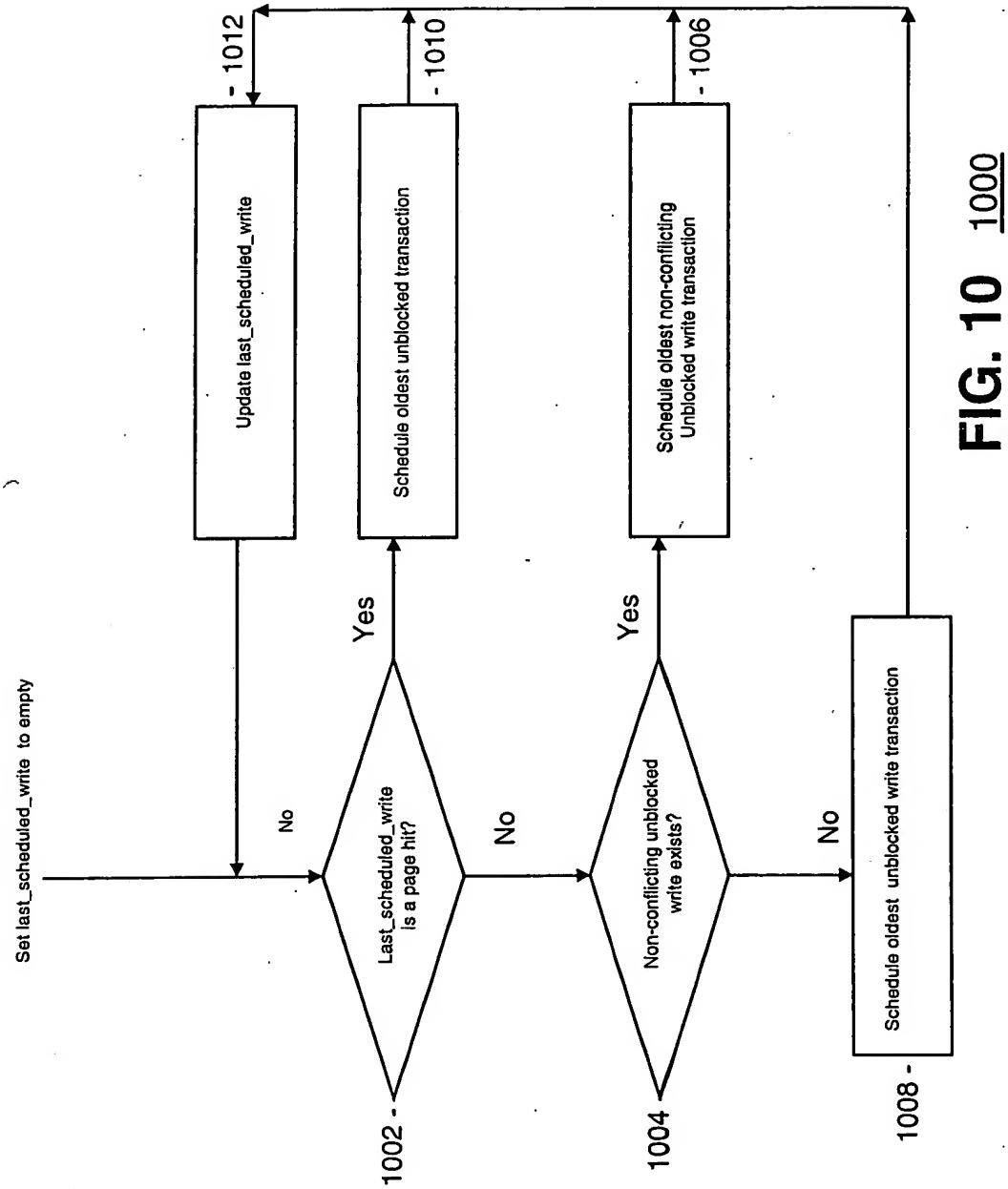

Set last_scheduled_write to empty
Last_scheduled_write is a page hit?
Yes
No
1002
Non-conflicting unblocked write exists?
Yes
No
1004
Schedule oldest unblocked write transaction
No
1008
Update last_scheduled_write
1012
Schedule oldest unblocked transaction
1010
Schedule oldest non-conflicting Unblocked write transaction
1006

FIG. 10 1000